Hierarchical Multi-label Text Classification: An Attention-based Recurrent Network Approach

Conference Paper · November 2019

DOI: 10.1145/3357384.3357885

9 authors, including:

Enhong Chen
University of Science and Technology of China
540 PUBLICATIONS **10,266** CITATIONS

SEE PROFILE

Hierarchical Multi-label Text Classification: An Attention-based Recurrent Network Approach

Wei Huang[1], Enhong Chen[1,*], Qi Liu[1], Yuying Chen[1,2], Zai Huang[1], Yang Liu[1], Zhou Zhao[3], Dan Zhang[4], Shijin Wang[4]
[1]School of Computer Science and Technology, University of Science and Technology of China
{cheneh,qiliuql}@ustc.edu.cn,{ustc0411,cyy33222,huangzai,ly0330}@mail.ustc.edu.cn
[2]Ant Financial Services Group, yuying.cyy@antfin.com
[3]School of Computer Science and Technology, Zhejiang University, zhaozhou@zju.edu.cn
[4]iFLYTEK Research, {danzhang, sjwang3}@iflytek.com

ABSTRACT

Hierarchical multi-label text classification (HMTC) is a fundamental but challenging task of numerous applications (e.g., patent annotation), where documents are assigned to multiple categories stored in a hierarchical structure. Categories at different levels of a document tend to have dependencies. However, the majority of prior studies for the HMTC task employ classifiers to either deal with all categories simultaneously or decompose the original problem into a set of flat multi-label classification subproblems, ignoring the associations between texts and the hierarchical structure and the dependencies among different levels of the hierarchical structure. To that end, in this paper, we propose a novel framework called *H*ierarchical *A*ttention-based *R*ecurrent *N*eural *N*etwork (HARNN) for classifying documents into the most relevant categories level by level via integrating texts and the hierarchical category structure. Specifically, we first apply a documentation representing layer for obtaining the representation of texts and the hierarchical structure. Then, we develop an hierarchical attention-based recurrent layer to model the dependencies among different levels of the hierarchical structure in a top-down fashion. Here, a hierarchical attention strategy is proposed to capture the associations between texts and the hierarchical structure. Finally, we design a hybrid method which is capable of predicting the categories of each level while classifying all categories in the entire hierarchical structure precisely. Extensive experimental results on two real-world datasets demonstrate the effectiveness and explanatory power of HARNN.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Natural language processing**;

KEYWORDS

Hierarchical Multi-label Text Classification; Attention Mechanism; Hierarchical Attention Networks

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357885

ACM Reference Format:
Wei Huang[1], Enhong Chen[1,*], Qi Liu[1], Yuying Chen[1,2], Zai Huang[1], Yang Liu[1], Zhou Zhao[3], Dan Zhang[4], Shijin Wang[4]. 2019. Hierarchical Multi-label Text Classification: An Attention-based Recurrent Network Approach. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357885

1 INTRODUCTION

Many real-world applications organize documents in a hierarchical structure, where classes are specialized into subclasses or grouped into superclasses [2, 32]. For example, an electronic document (e.g., web-page, patent and e-mail) is associated with multiple categories and all these categories are stored hierarchically in a tree or a Direct Acyclic Graph (DAG) [16]. Figure 1 shows a toy example of a patent document with a four-level hierarchical category structure, the top part is a detailed text description of the patent, and its corresponding hierarchical categories with a tree graph representation are shown below. The root node is considered as level 0, and the parent nodes represent more general than the child nodes. It is an elegant way to show the characteristics of data and a multi-dimensional perspective to tackle the classification problem via the hierarchical structure. Thus, this kind of problem, known as *hierarchical multi-label text classification* (HMTC), has aroused widespread attraction in both the industry and the academia [11, 14, 22, 25].

In the literature, there are many efforts for HMTC problem. Initially, the flat-based methods (e.g., Naive Bayes) have been proposed to predict only the categories of the last level by reducing the HMTC problem into a flat multi-label problem [12]. Unfortunately, these simple approaches ignore the hierarchical category structure information (e.g., *nuclear physics* is the subclass of *physics* in Figure 1). To tackle the problem above, some works have taken the hierarchical structure into consideration [2, 6, 10, 24, 27], which can be further compartmentalized into two approaches: (1) Local approaches generate hierarchical classifiers and each classifier is responsible for predicting either corresponding categories [2] or corresponding category levels [27]; (2) Global approaches gather all the levels of categories together and predict them with a single classifier [10]. However, these studies only focus on either the local regions or the overall structure of the category hierarchy, while ignoring the dependencies among different levels of the hierarchical structure.

In summary, there are still many unique challenges inherent in designing an effective HMTC solution. First, in a text classification task, different parts of the document are associated with different

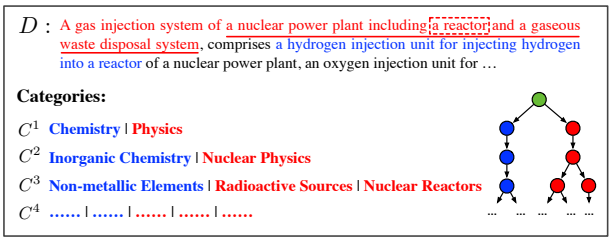


Figure 1: A toy example of a patent document in HTMC problem.

categories. For instance, in Figure 1, the "red" words of document D focus more on *physics* while the "blue" words concentrate more on *chemistry* in C^1 (level-1 category), and the words on the "red" underline further describe *nuclear physics* in C^2 (level-2 category). Therefore, when understanding the semantic of each document, it is necessary to capture these associations between texts and the hierarchical structure. Second, there are dependencies among different levels in the hierarchical category structure (i.e., the determination of a category not only is influenced by its parent category but also will affect its child categories). As shown in Figure 1, document D concentrates on the *nuclear physics* in C^2 since its parent category *physics* in C^1 is focused. And *nuclear reactors* in C^3 is the subclass of *nuclear physics*, which should also be highly heeded. Thus, it is also critical to classify level by level via considering the dependencies among different levels of the hierarchical structure. Third, when assigning the document into hierarchical categories, not only the local regions but also the overall structure of the category hierarchy should be considered. Hence, how to predict the categories of each level while classifying all categories in the entire hierarchical structure precisely is a nontrivial problem.

To address the challenges mentioned above, in this paper, we propose a novel framework named Hierarchical Attention-based Recurrent Neural Network (HARNN), which can automatically annotate a document with the most relevant categories level by level. Specifically, given the documents and the whole hierarchical category structure, we first apply a Documentation Representing Layer (DRL) for obtaining the representation of texts and the hierarchical structure. Then, we devise an Hierarchical Attention-based Recurrent Layer (HARL) to model the dependencies among different levels by leveraging the hierarchical structure gradually in a top-down fashion. To be specific, at each level, the attention mechanism qualifies the contribution of each text word to each category, and the text-category association information will affect the next category level. Throughout the recurrence, the Hierarchical Attention-based Memory (HAM) unit we designed can model the dependencies among different levels of the hierarchical structure. After that, we utilize a Hybrid Predicting Layer (HPL) for combining the predictions of each level in the category hierarchy and the overall hierarchical structure. Thus, the proposed model is capable of predicting the categories of each level while classifying all categories in the entire hierarchical structure precisely. Finally, extensive experiments on two large-scale real-world datasets demonstrate the effectiveness and explanatory power of our model. To the best of our knowledge, this is the first comprehensive attempt to employ hierarchical attentive neural networks for HMTC problem.

2 RELATED WORK

Generally, the related works can be grouped into two research aspects, i.e., studies on hierarchical multi-label text classification and attention mechanism.

2.1 Studies on HMTC

There have been several efforts for HMTC in the literature. Initially, flat-based methods were widely used in HMTC task, such as Decision Tree (DT) and Naive Bayes (NB) [12], which learned discriminant classifiers only for the categories of the last level in the category hierarchy. However, the hierarchical structure information was ignored in these approaches, which caused the inefficiency. Thus, some hierarchical approaches were proposed by taking the hierarchical structure information into consideration and could be compartmentalized into local and global approaches according to the strategy adopted. Regarding local approaches, Cesa-Bianchi et al. [8] proposed a classification method using hierarchical SVM, where SVM learning was applied to a category only if its parent category was labeled as positive. Afterwards, a large margin method was proposed by Rousu et al. [26] to calculate the maximum structural margin for the output categories. As for global approaches, Vens et al. [31] developed a tree-based approach called Clus-HMC to employ a single decision tree to deal with the entire hierarchical category structure. Recently, neural networks have been utilized for HMTC problems and have shown its effectiveness. Cerri et al. [7] attempted to use HMC-LMLP for incrementally training a set of neural networks and each being responsible for predicting the categories in a given level. Borges et al. [4] proposed a global approach based on a competitive artificial neural network to predict all categories in the hierarchical structure.

Nevertheless, the above works mainly focus on either the local regions or the overall structure of the category hierarchy. Moreover, they neglect the dependencies among the different levels of the hierarchical structure, which leads to error propagation and well-known class-membership inconsistency [28]. Along this line, methods combing the advantages of local and global approaches have been applied [33] in many domains (e.g., text classification, protein function prediction), where a neural network HMCN was proposed for integrating the predictions of each level in the category hierarchy and the overall hierarchical structure. Unfortunately, they failed to capture the associations between texts and the hierarchical structure, which are important for understanding the semantics of each document.

2.2 Attention Mechanism

In our framework, one of the most significant steps is to reveal the associations between texts and each category in the hierarchical structure from top to down and level by level, which is related to the attention mechanism [37]. In text classification, attention mechanism is a powerful approach to highlight different parts of the text semantic representation by assigning different weights. For example, Huang et al. [18] used attention mechanism on the top of a CNN model to introduce an extra source of information for guiding the extraction of the sentence embedding. Tao et al. [30] utilized an attention method which created a weighted vector representation by using the encodings of either RNNs hidden states. Lin et al. [20]

designed a self-attention mechanism for the sequential models (e.g., RNN) to replace the max pooling or averaging step, which enabled their model pay attention to the different aspects of the sentence.

In the aforementioned works, the attention weights are usually calculated by the correspondence between texts and each category in particular levels, which treats the different levels of the hierarchical structure independently thereby ignores the dependencies among the different levels. In this work, we propose a novel hierarchical attention recurrent structure to capture the associations between texts and each category gradually from top to down, which integrates the dependencies among different levels. Specifically, in the hierarchical attention mechanism, the attention weights of texts and each category in a level not only are influenced by its previous level but also will affect the next level.

3 PRELIMINARIES

3.1 Problem Definition

In HMTC problem, there are a set of documents (e.g., patent) and each document contains the text description and its expected categories which are organized in a hierarchical structure. Before defining the HMTC problem, we start with a detailed description of the hierarchical structure and documents.

Definition 3.1. (**Hierarchical Structure** γ). Given the defined possible categories in H hierarchical levels $\mathbb{C} = (C^1, C^2, \ldots, C^H)$, where $C^i = \{c_1, c_2, \ldots\} \in \{0,1\}^{|C^i|}$ is the set of possible categories in the i-th hierarchical level and $|C^i|$ is the number of categories in the i-th hierarchical level and K is the total number of categories. We define the hierarchical category structure γ over $\mathbb{C}$ as a partial order set $(\mathbb{C}, \prec)$. $\prec$ is a partial order representing the PARENT-OF relationship, which is asymmetric, anti-reflexive and transitive [34]:

- $\forall c_x \in C^i, c_y \in C^j, c_x \prec c_y$ then $c_y \nprec c_x$
- $\forall c_x \in C^i, c_x \nprec c_x$
- $\forall c_x \in C^i, c_y \in C^j, c_z \in C^k$, if $c_x \prec c_y$ and $c_y \prec c_z$ then $c_x \prec c_z$

A set of M documents with expected hierarchical categories can be denoted as $\mathcal{X} = \{(D_1, L_1), (D_2, L_2) \ldots, (D_M, L_M)\}$, where $D_i = \{w_1, w_2, \ldots, w_N\}$ can usually be represented as a sequence of N words, and $L_i = \{\ell_1, \ell_2, \ldots, \ell_H\}$ is the set of excepted hierarchical categories assigned to the document D_i, with $\ell_i \subset C^i$ in the corresponding hierarchical structure γ. Then, the HMTC problem can be formulated as:

Definition 3.2. (**HMTC Problem**). Given a set of documents and the corresponding hierarchical category structure, our goal is to integrate the document texts D and the corresponding hierarchical category structure γ to learn a classification model Ω, which can be used to predict the hierarchical categories L for documents:

$$\Omega(D, \gamma, \Theta) \rightarrow L, \tag{1}$$

where Θ is the parameters of Ω.

4 MODEL ARCHITECTURE

In this section, we will introduce the technical details of HARNN framework. As shown in Figure 2, HARNN mainly contains three parts, i.e., Documentation Representing Layer (DRL), Hierarchical Attention-based Recurrent Layer (HARL) and Hybrid Predicting Layer (HPL). Specifically, we utilize DRL to obtain the unified representation of each document text and the hierarchical category structure. Then, we design HARL to model the dependencies among different levels via capturing the associations between texts and each category of the hierarchical structure in a top-down fashion. Finally, HPL is applied to predict the hierarchical categories of documents.

4.1 Documentation Representing Layer

In the first stage of HARNN, DRL aims to generate the unified representation of the document text and the hierarchical category structure. To this end, we first apply an Embedding Layer to encode text and the hierarchical category structure. Then a Bi-LSTM Layer is utilized to enhance the encodings of text semantic representation.

Embedding Layer is used to encode the text and the hierarchical category structure. DRL receives the text tokens of a document D and the hierarchical category structure γ as input, as shown in Figure 2. Intuitively, the document text D is formalized as a sequence of N words $D = (w_1, w_2, \ldots, w_N)$, where $w_i \in \mathbb{R}^k$ is initialized by a k-dimensional pre-trained word embedding with *Word2vec* [23]. Like *Word2vec* operation in text tokens, we embed the hierarchical category structure γ into a matrix $\boldsymbol{S} = (S^1, S^2, \ldots, S^H)$ for training, where $S^i \in \mathbb{R}^{|C^i| \times d_a}$ is a randomly initialized matrix which represents the embedding of the i-th hierarchical category level with the d_a-dimension. After the initialization from Embedding Layer, we apply the subsequent Bi-LSTM Layer to enhance the semantic representations of the document text.

Bi-LSTM Layer targets at enhancing the encodings of the text semantic representation. For the text tokens in D, we utilize a Bi-LSTM architecture, which is a variant of the traditional LSTM architecture [15, 17]. Bi-LSTM can learn not only long-range dependencies across the input sequence but also context information from forward and backward simultaneously, which is beneficial for enhancing the encodings of the text semantic representation. Specifically, the input of the Bi-LSTM network is a sequence $D = (w_1, w_2, \ldots, w_N)$, and the hidden vector of a Bi-LSTM is calculated as follows:

$$\begin{aligned} \overrightarrow{h_n} &= LSTM(\overrightarrow{h}_{n-1}, w_n), \\ \overleftarrow{h_n} &= LSTM(\overleftarrow{h}_{n+1}, w_n), \\ h_n &= [\overrightarrow{h_n}, \overleftarrow{h_n}], \end{aligned} \tag{2}$$

where $\overrightarrow{h_n}$ and $\overleftarrow{h_n}$ are the forward hidden vector and the backward hidden vector respectively at the n-th word step in the Bi-LSTM. And $h_n \in \mathbb{R}^{2u}$ is the hidden output of Bi-LSTM at the n-th word step, which is the concatenation of $\overrightarrow{h_n}$ and $\overleftarrow{h_n}$, where u is the hidden unit number of each unidirectional LSTM.

For simplicity, we note all h_n as $\boldsymbol{V} = \{h_1, h_2, \ldots, h_N\} \in \mathbb{R}^{N \times 2u}$. After that, to obtain the whole semantic representation of the document D, we exploit the word-wise average pooling operation to merge N words contextual representations into an average embedding $\widetilde{V}$ as $\widetilde{V} = avg(h_1, h_2, \ldots, h_N) \in \mathbb{R}^{2u}$.

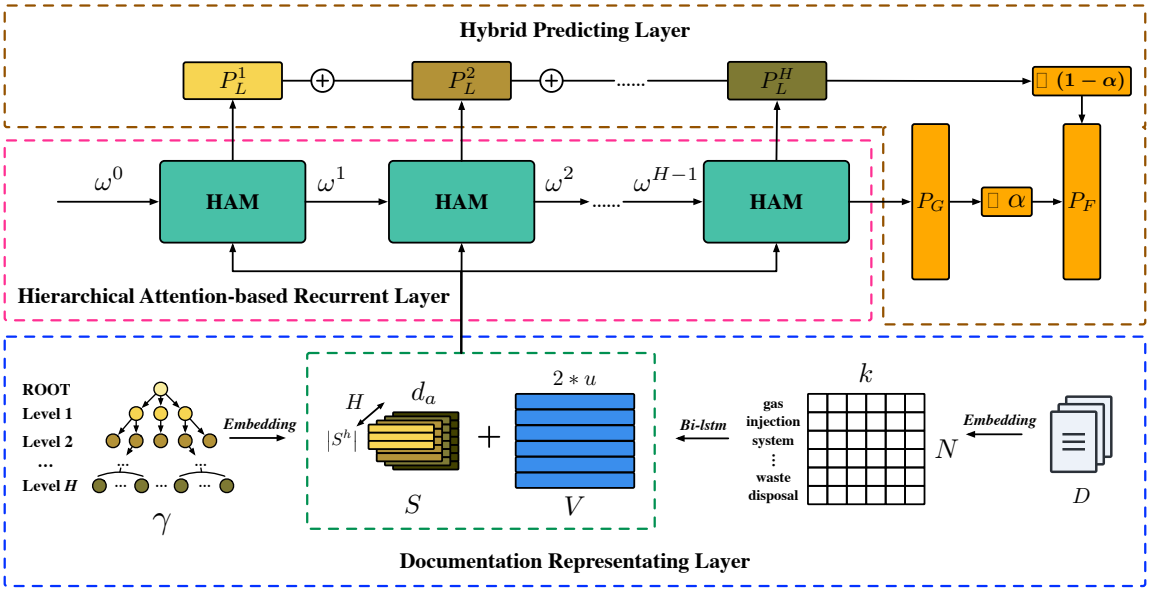


Figure 2: The HARNN framework takes a given document text D and the corresponding hierarchical category structure γ as inputs, and outputs the hierarchical categories L for the given document.

4.2 Hierarchical Attention-based Recurrent Layer

After obtaining the unified representation of the text and the hierarchical category structure, we propose HARL, a recurrent architecture, to model the dependencies among different levels by leveraging the hierarchical structure gradually in a top-down fashion. At each category level, the core repeating unit, i.e., Hierarchical Attention-based Memory (HAM) unit is explicitly designed to capture the associations between texts and categories. Meanwhile, it can transfer the corresponding hierarchical semantic representation to the next layer. This hierarchical semantic representation is consistent with the human reading habit that people usually understand the concept of a document from shallow to deep. As shown in Figure 1, the "red" words in D pay more attention to *physics* concept in C^1 (level-1 category) while the words on "red" underline focus more on *nuclear physics* concept in C^2. Obviously, *nuclear physics* is the subclass of *physics*. Therefore, it is necessary to qualify the contributions of the text semantic representation to the category hierarchy from top to down and learn the attention representations for each category level.

Figure 3 shows the details of an HAM unit. An HAM unit mainly consists of three components, i.e., Text-Category Attention (TCA), Class Prediction Module (CPM) and Class Dependency Module (CDM). Specifically, TCA first captures the associations between texts and categories of the hierarchical structure. Based on the text-category associations, CPM generates the unified representation and predictions of the corresponding category level. Next, CDM is utilized to model the dependencies among different levels of the hierarchical structure. For the h-th category level in γ, the input to corresponding HAM unit includes three parts, i.e., the whole text semantic representation $\boldsymbol{V}$, the corresponding category level representation S^h, and the information of previous level $\boldsymbol{\omega^{h-1}}$. The

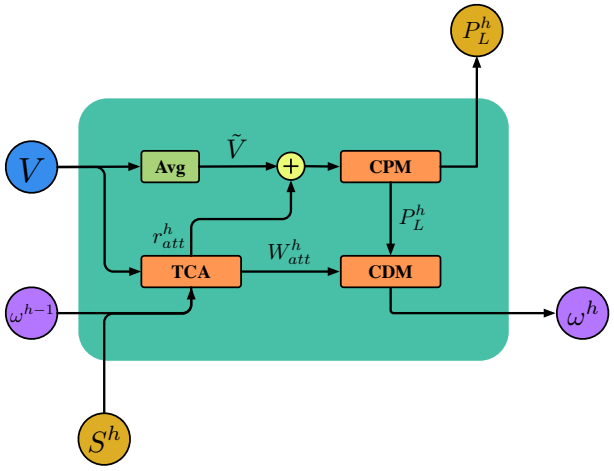


Figure 3: Hierarchical attention-based memory unit.

unified representation A_L^h and information $\boldsymbol{\omega^h}$ at the h-th level are updated as following formulas:

$$\begin{aligned} \widetilde{V} &= avg(\boldsymbol{V}), \\ r_{att}^h, W_{att}^h &= TCA(\boldsymbol{\omega^{h-1}}, \boldsymbol{V}, S^h), \\ P_L^h, A_L^h &= CPM([\widetilde{V} \oplus r_{att}^h]), \\ \boldsymbol{\omega^h} &= CDM(W_{att}^h, P_L^h), \end{aligned} \tag{3}$$

where $\widetilde{V}$ is the whole text semantic representation, r_{att}^h is the associated text-category representation of h-th level, W_{att}^h is the text-category attention matrix at h-th level, P_L^h denotes the predicted probability vector of h-th level, $\oplus$ denotes vector concatenation operation and $avg()$ is the average pooling operation. Note that all

elements in ω^0 are initialized as 1.0 since the level 0 is the root of the hierarchical structure which contains no extra information.

Next, we introduce each component of the HAM unit.

Text-Category Attention targets at capturing the associations among texts and categories, and outputs the associated text-category representation r^h_{att} and text-category attention matrix W^h_{att} at h-th category level. As shown in Figure 4, given the information of previous category level $\boldsymbol{\omega^{h-1}} \in \mathbb{R}^{N\times 2u}$, we integrate it with the whole text semantic representation $\boldsymbol{V} \in \mathbb{R}^{N\times 2u}$ to generate the V_h:

$$V_h = \boldsymbol{\omega^{h-1}} \otimes \boldsymbol{V}, \tag{4}$$

where $V_h \in \mathbb{R}^{N\times 2u}$ denotes the representation which introduces the hierarchical information of previous category level, and $\otimes$ denotes entry-wise product operation.

Inspired by [20], the document text is usually formed by multiple components which focus on different aspects, especially for the long sentences. For instance, in Figure 1, the "red" words of document D focus more on *physics* while the "blue" words concentrate more on *chemistry*. Thus, for the h-th category level, we need $|C^h|$ aspects for focusing on different categories to represent the overall semantic of the whole sentence. We use $S^h \in \mathbb{R}^{|C^h|\times d_a}$ which is the embedding of h-th category level to perform $|C^h|$ different categories of attention. Formally,

$$\begin{aligned} O_h &= tanh(\boldsymbol{W_s^h} \cdot V_h^T), \\ W^h_{att} &= softmax(S^h \cdot O_h), \end{aligned} \tag{5}$$

where $\boldsymbol{W_s^h} \in \mathbb{R}^{d_a\times 2u}$ is a randomly initialized weight matrix, O_h denotes the activations which can be viewed as one MLP without bias, d_a is a hyperparameter we can set arbitrarily, the *softmax*() ensures all the computed weights sum up to 1 for each category. And $W^h_{att} = (W^h_1, W^h_2, \ldots, W^h_{|C^h|}) \in \mathbb{R}^{|C^h|\times N}$ is the annotation matrix, where $W^h_i \in \mathbb{R}^N$ represents the attention score of the text with i-th category in h-th level after normalization and each element in this vector represents the contribution of each word token on its corresponding position contributes to this i-th category.

Then, we compute $|C^h|$ weighted sums by multiplying the annotation matrix W^h_{att} and the text semantic representation V_h. The resulting matrix $M_h \in \mathbb{R}^{|C^h|\times 2u}$ is the associated text-category representation with each category in h-th level. And $r^h_{att} \in \mathbb{R}^{2u}$ which is the associated text-category representation for whole h-th level can be modeled as a vector by averaging M_h in category-dims:

$$\begin{aligned} M_h &= W^h_{att} \cdot V_h, \\ r^h_{att} &= avg(M_h). \end{aligned} \tag{6}$$

With the help of Text-Category Attention, we can get the associated text-category representation r^h_{att} and the annotation matrix W^h_{att} for the h-th category level.

Class Prediction Module aims at generating the unified representation and predicting the categories for each level by integrating the original text semantic representation and the associated text-category representation which introduces the information of its previous level. Formally, let A^h_L denote the representation of h-th level and A^h_L is given by:

$$A^h_L = \varphi(W^h_T \cdot [\widetilde{V} \oplus r^h_{att}] + b^h_T), \tag{7}$$

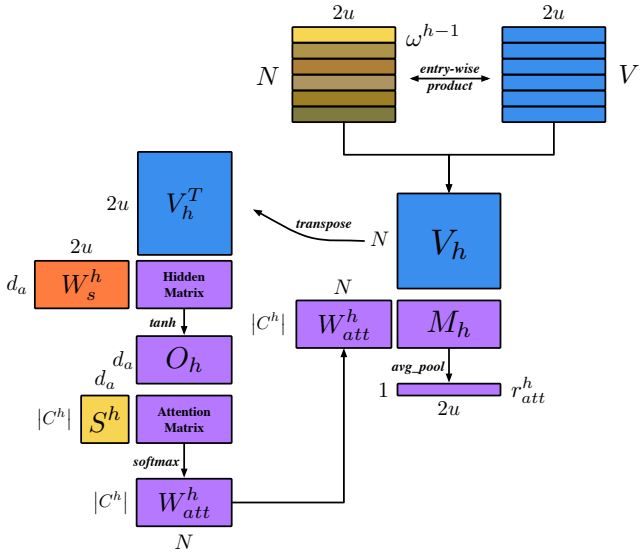


Figure 4: Text-Category Attention.

where $W^h_T \in \mathbb{R}^{v\times 4u}$ is a randomly initialized weight matrix and $b^h_T \in \mathbb{R}^{v\times 1}$ is the corresponding bias vector, φ is a non-linear activation function (e.g., RELU). Then, the local prediction of h-th category level P^h_L is calculated by:

$$P^h_L = \sigma(W^h_L \cdot A^h_L + b^h_L), \tag{8}$$

where $W^h_L \in \mathbb{R}^{|C^h|\times v}$ is the weighted matrix that connects the activations of h-th level with $|C^h|$ output units, $b^h_L \in \mathbb{R}^{|C^h|\times 1}$ is the corresponding bias vector, and σ is the sigmoid activation.

Class Dependency Module is utilized to model the dependencies between different levels of the hierarchical structure by keeping the hierarchical information for each level. For the h-th category level, different categories have different contributions to the prediction, which can be used as the trade-off parameter for revising text-category attention matrix. Therefore, we first utilize an entry-wise product operation to combine the predicted category values P^h_L and the text-category attention matrix W^h_{att} to generate the weighted text-category attention matrix K^h:

$$K^h = broadcast(P^h_L) \otimes W^h_{att}, \tag{9}$$

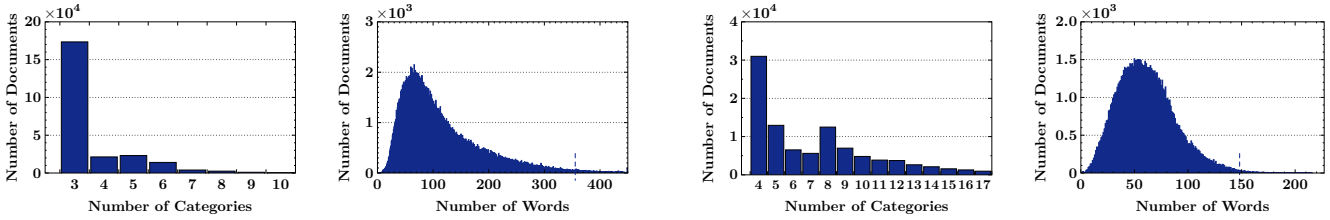
where $K^h = (k^h_1, \ldots, k^h_{|C^h|}) \in \mathbb{R}^{|C^h|\times N}$ is the weighted text-category attention matrix of h-th level considering the different categories should have the different weight, k^h_i denotes weighted attention score of the i-th category with the text semantic representation in h-th level, and *broadcast*() is the process of making matrixes with different shapes have compatible shapes for arithmetic operations (i.e., entry-wise product).

Then we exploit the category-wise average pooling operation to merge $|C^h|$ categories into an average representation $\widetilde{K}^h$:

$$\widetilde{K}^h = avg(K^h), \tag{10}$$

where $\widetilde{K}^h \in \mathbb{R}^N$ is the weighted attention vector of h-th level which holds the attention associations of the whole category level with the text.

Next, we broadcast the average representation into $\boldsymbol{\omega^h}$:



(a) Education (b) Patent

Figure 5: Number distributions of observed records.

$$\boldsymbol{\omega^h} = broadcast(\widetilde{K}^h), \tag{11}$$

where $\boldsymbol{\omega^h} = (\omega_1^h, \omega_2^h, \ldots, \omega_N^h) \in \mathbb{R}^{N\times 2u}$ is a matrix holding the hierarchy information and the associations between the text and the whole category level at h-th level, and $\omega_i^h \in \mathbb{R}^{2u}$ measures the association weights between the whole previous category level and the i-th word token in D.

Finally, we bring $\boldsymbol{\omega^h}$ to the next category level since the information of each category level not only is influenced by the previous level but also will affect the next level.

4.3 Hybrid Predicting Layer

Throughout HARL, we can obtain the unified representation A_L^h and the predictions P_L^h of each category level. It is important to predict all categories in the entire hierarchy, and predicting the categories of each level in HMTC problem is also critical. Thus, we apply a Hybrid Predicting Layer, a hybrid prediction method to generate the final predictions by considering both local and global prediction information. Specifically, we note all the A_L^h as $A_L = \{A_L^1, A_L^2, \ldots, A_L^H\} \in \mathbb{R}^{H\times v}$, and then exploit the level-wise average pooling operation to merge H levels representations into a global embedding $\widetilde{A}_L = avg(A_L^1, A_L^2, \ldots, A_L^H) \in \mathbb{R}^v$. Let P_G denote the global predictions of the entire category hierarchy and P_G is given by:

$$\begin{aligned} A_G &= \varphi(W_G \cdot \widetilde{A}_L + b_G), \\ P_G &= \sigma(W_M \cdot A_G + b_M), \end{aligned} \tag{12}$$

where $W_G \in \mathbb{R}^{v\times v}$, $W_M \in \mathbb{R}^{K\times v}$ are randomly initialized weight matries, $b_G \in \mathbb{R}^{v\times 1}$ and $b_M \in \mathbb{R}^{K\times 1}$ are corresponding bias vectors. P_G is a continuous vector and each element in this vector P_G^i denotes the probability $P(c_i|x)$ for $c_i \in \mathbb{C}$.

In order to integrate both local and global predictions, the final predictions P_F is calculated by:

$$P_F = (1-\alpha)\cdot(P_L^1 \oplus P_L^2 \oplus, \ldots, P_L^H) + \alpha \cdot P_G, \tag{13}$$

where $\alpha \in [0, 1]$ is a balance parameter, which is used to regulate the trade-off regarding the local and global predictions. We set $\alpha = 0.5$ as the default option in order to give equal importance to both local and global predictions.

4.4 Loss Function

In this subsection, we specify a loss function which is the sum of the local and global loss functions for each document to train HARNN. In the training stage, for each document D, the local ($\mathcal{L}_L$) and global ($\mathcal{L}_G$) loss are calculated as:

Table 1: The statistics of the datasets

Statistics	Education	Patent
# instances	240,309	100,000
# hierarchical levels	3	4
# categories in level-1	18	9
# categories in level-2	75	128
# categories in level-3	366	661
# categories in level-4	-	8364
# total categories	459	9162
# average categories per instance in level-1	1.07	1.46
# average categories per instance in level-2	1.21	1.62
# average categories per instance in level-3	1.36	1.82
# average categories per instance in level-4	-	2.78
# average categories per instance	3.64	7.67

$$\begin{aligned} \mathcal{L}_L &= \sum_{h=1}^{H}[\varepsilon(P_L^h, Y_L^h)], \\ \mathcal{L}_G &= \varepsilon(P_G, Y_G), \end{aligned} \tag{14}$$

where Y_G is the binary label vector (expected output) containing all categories of the hierarchical structure and Y_L^h is the binary label vector (expected output) containing only the categories of the h-th level. Since categories are not mutually exclusive, we use $\varepsilon(\cdot,\cdot)$ the binary cross-entropy to minimize the local and global loss function of a document. The final loss function we optimize is thus given by:

$$\mathcal{L}(\Theta) = \mathcal{L}_L + \mathcal{L}_G + \lambda_\Theta ||\Theta||^2, \tag{15}$$

where Θ denotes all parameters of HARNN and λ_Θ is the regularization hyperparameter. In this way, we can learn HARNN by directly minimizing the loss function $\mathcal{L}(\Theta)$ using Adam [19].

5 EXPERIMENTS

In this section, we first introduce the datasets and our experimental setups. Then, we conduct extensive experiments on HARNN model and the other state-of-the-art methods to answer the following questions:

- **RQ1**: How does HARNN perform in predicting total categories of the entire hierarchical structure compared to the state-of-the-art methods?
- **RQ2**: How does HARNN perform in predicting categories of each level in the hierarchical structure compared to the other hierarchical approaches?
- **RQ3**: How does the trade-off coefficient (α) between local and global predictions influences the performance ?
- **RQ4**: Is the proposed hierarchical attention mechanism helpful and explanatory in HTMC problem?

Table 2: Performance comparison on the HMTC task.

(a) Education

Baseline	Metrics			
	Precision	Recall	micro-F1	$AU(\overline{PRC})$
Clus-HMC	0.806	0.704	0.752	0.685
HMC-LMLP	0.841	0.732	0.783	0.861
HMCN-R	0.844	0.733	0.785	0.867
HMCN-F	0.843	0.739	0.787	0.865
HARNN-LG	0.853	0.744	0.795	0.876
HARNN-LH	0.860	0.740	0.795	0.879
HARNN-GH	0.845	0.747	0.793	0.874
HARNN	**0.860**	**0.767**	**0.811**	**0.893**

(b) Patent

Baseline	Metrics			
	Precision	Recall	micro-F1	$AU(\overline{PRC})$
Clus-HMC	0.419	0.345	0.379	0.145
HMC-LMLP	0.692	0.380	0.490	0.526
HMCN-R	0.684	0.395	0.501	0.528
HMCN-F	0.704	0.376	0.491	0.524
HARNN-LG	0.738	0.420	0.535	0.579
HARNN-LH	0.733	0.418	0.532	0.578
HARNN-GH	0.740	0.405	0.523	0.572
HARNN	**0.742**	**0.425**	**0.541**	**0.583**

5.1 Dataset Description

We conduct experiments on two real-world datasets: *educational exercises dataset* and *patent documents dataset*, which contain document text records and hierarchical category structures.

Education Dataset is collected from an online education system which provides a series of exercise-based applications for high school students in China. The dataset contains 240,309 real-world math exercises and each math exercise contains multiple categories which are organized into a tree structure with three levels.

Patent Dataset is collected from the USPTO[1] which is a patent system grating U.S. patents. The dataset includes 100,000 real-world granted US patents and each patent document includes textual information (e.g., title, abstract) and multiple hierarchical categories. Figure 1 shows a toy example of a patent document with a four-level hierarchical category structure.

Table 1 and Figure 5 show the basic statistics and the documentation distribution of the two datasets respectively, and we can observe that: (1) Education dataset has total 459 categories which are organized into a three-level hierarchical structure while Patent dataset has total 9,162 categories with a four-level hierarchical structure; (2) Each document consists of about 3.64 and 7.67 categories in Education and Patent dataset respectively. These categories are distributed in different levels, and each level basically averages one or two categories; (3) About 95% documents of Education dataset contain fewer than 350 words, while 95% documents of Patent dataset contain fewer than 150 words.

The average number of category per instance is quite small compared to the total number of category in the whole hierarchical structure. Thus, when predicting the multiple categories for each document, it is critical to integrate the dependencies between different levels of the hierarchical structure to reduce the influence of the other irrelevant category information. Moreover, it is also necessary to capture the associations between texts and the hierarchical structure, especially for the long sentences, which has been fully discussed in Section 4.2.

5.2 Experimental Setup

For validating the effectiveness of HARNN, we employ 10-fold cross-validation on labeled documents in two datasets, where one of ten folds is targeted to construct the testing set and the rest for the training set.

Word Embedding Pre-training. We use the *Word2vec* tool [23] with a dimension (k) 100 to generate the word embedding for each word in a document, which is trained on texts of documents in the datasets.

HARNN Setting.[2] We set the size (d_a) of embedding representations for categories as 200, the dimension (u) of hidden states in Bi-LSTM as 256, the dimension (v) of the all full-connected layer units as 512 and the parameter (α) of local & global information trade-off regulation as 0.5.

Training Details. In HARNN, we set the maximum length N of words in a document as 350 in Education dataset while 150 in Patent dataset (zero padded when necessary) according to our observation in Figure 5. We initialize parameters in HARNN with a truncated normal distribution and the standard deviation is set as 0.1. For training HARNN, we use the Adam optimizer [19] with a learning rate of 1×10^{-3}, and set the mini-batches as 256, $\lambda_\Theta = 0.00004$, $\beta = 0.1$ in Eq. (15). All parameters of HARNN can be tuned during the training process. Note that all the fully-connected layers comprise 512 ReLU neurons. We also use dropout [29] with a drop-rate of 0.5 to prevent overfitting and gradient clipping to avoid the gradient explosion problem.

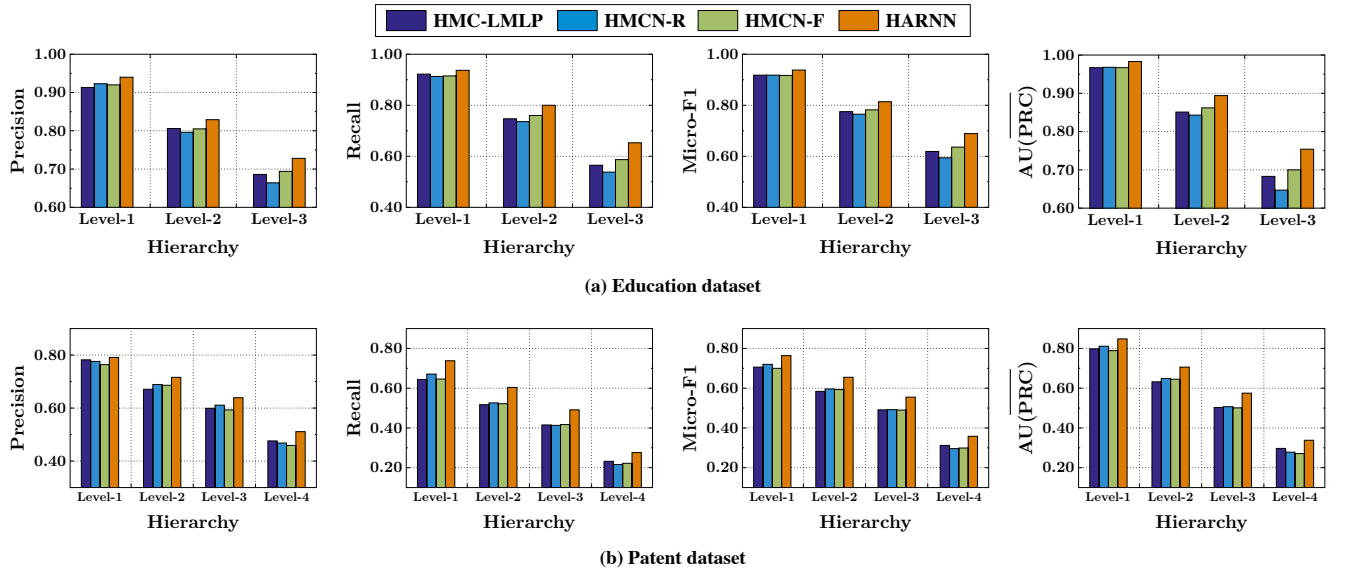
5.3 Baseline Approaches

In order to demonstrate the effectiveness of HARNN, we compare it with several HMTC methods including four state-of-the-art models and some variants of HARNN:

- **Clus-HMC** [31]: Clus-HMC is a global approach that builds a single decision tree to classify all categories simultaneously.
- **HMC-LMLP** [7]: HMC-LMLP is a local approach which trains a set of neural networks and each is responsible for the prediction of the categories belonging to a given level.
- **HMCN-F** [33]: HMCN-F is a feed-forward hybrid approach which combines the prediction of each level in the category hierarchy and the overall hierarchy structure.
- **HMCN-R** [33]: HMCN-R is a recurrent hybrid approach which combines the prediction of each level in the category hierarchy and the overall hierarchy structure.

[1] https://www.uspto.gov/

[2] The code is available at https://github.com/RandolphVI/Hierarchical-Multi-Label-Text-Classification



(a) Education dataset

(b) Patent dataset

Figure 6: Performance on different levels in hierarchy.

The variants of HARNN are listed as follows:

- **HARNN-LG**: HARNN-LG is a variant of HARNN without considering the dependencies between different levels of the hierarchical structure.
- **HARNN-LH**: HARNN-LH is a variant of HARNN without considering the overall hierarchical structure information.
- **HARNN-GH**: HARNN-GH is a variant of HARNN without considering the information of the local regions in the hierarchical structure.

Concretely, the chosen baselines can be categorized into local approach (HMC-LMLP), global approach (Clus-HMC) and hybrid approachs (HMCN-F, HMCN-R). All baselines ignore the associations between texts and the hierarchical structure. Note that we design three variants of HARNN to highlight the effectiveness of our proposed HAM unit and hybrid prediction method. In the following experiments, all methods are implemented by TensorFlow [1], and all the experiments are conducted on a Linux server with four 2.0GHz Intel Xeon E5-2620 CPUs and a Tesla K80 GPU. For fair comparisons, all parameters in these baselines are tuned to achieve the best performance.

5.4 Evaluation Metrics

5.4.1 ***Threshold based evaluation***. In HMTC task, we first adopt three widely used evaluation metrics [3, 5]: *precision*, *recall* and *F1* measure. Given a category $i \in \mathbb{C}$, let TP_i, FP_i, FN_i, be the number of true positives, false positives, false negatives, respectively. The precision and recall for the whole output hierarchical structure are then defined as [31]:

$$P = \frac{\sum_{i \in \mathbb{C}} TP_i}{\sum_{i \in \mathbb{C}} TP_i + \sum_{i \in \mathbb{C}} FP_i}, \quad R = \frac{\sum_{i \in \mathbb{C}} TP_i}{\sum_{i \in \mathbb{C}} TP_i + \sum_{i \in \mathbb{C}} FN_i}. \quad (16)$$

And we also evaluate the performance using $micro - F_1$ [13] which is the combination of precision and recall. The $micro - F_1$ measures the classification effectiveness by counting the true positives, false negatives and false positives globally, which takes class imbalance into account [36]. For the three threshold based evaluation metrics, the larger, the better.

5.4.2 ***Area under the average PR curve***. The outputs of our model HARNN and the other baselines are probability values of each category in the entire hierarchical structure. Hence, the final predictions are generated after thresholding those probabilities. The choice of an optimal threshold is difficult and often subjective, thus we follow the trend of HMTC research [7, 33] to employ *Area Under the Average Precision-Recall* Curve ($AU(\overline{PRC})$) [9] for avoiding choosing thresholds. The higher the $AU(\overline{PRC})$ of a method is, the better its predictive performance is.

5.5 Experimental Results

5.5.1 ***Performance Comparison (RQ1)***. To demonstrate practical significance of our proposed model, we compare HARNN with all the baselines on HMTC task. The results of all methods on both Education and Patent datasets are shown in Table 2. From the results, we can get several observations:

(1) HARNN achieves the best performance on all evaluation metrics in two datasets, which indicates that HARNN is more capable for HMTC tasks with an advantage of tackling hierarchical category structure effectively and accurately.
(2) All the variants of HARNN perform better than HMCN-F and HMCN-R on all evaluation metrics in two datasets. The reason is that the variants of HARNN could capture the associations between texts and the hierarchical structure via TCA while HMCN-F and HMCN-R cannot.
(3) HARNN beats HARNN-LG by additionally leveraging the HAM unit which considers the dependencies among different levels of the hierarchical structure. And HARNN performs better than HARNN-LH and HARNN-GH by combining the information

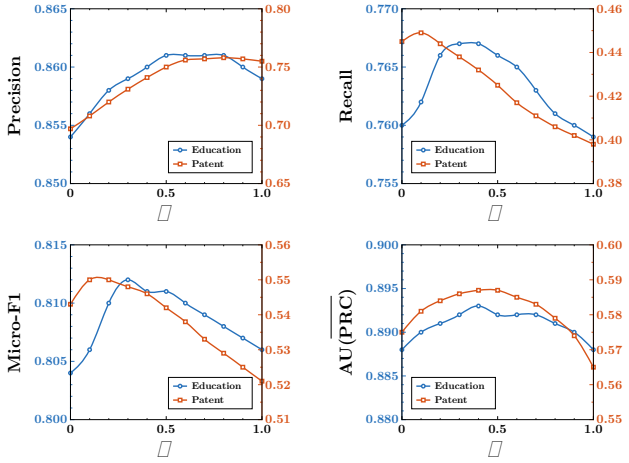


Figure 7: Model performance with different α.

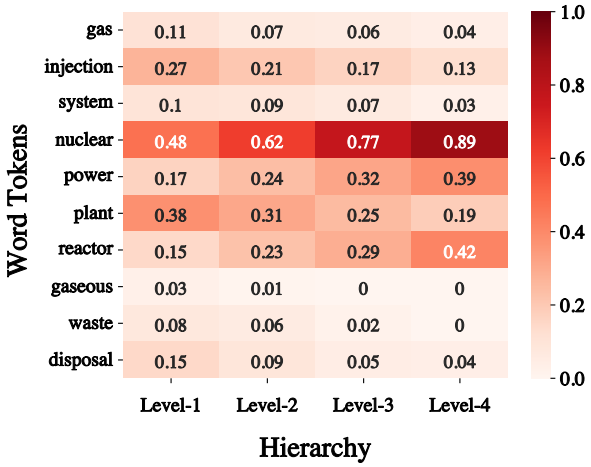


Figure 8: Illustration of learned attention weights.

of the local regions in the category hierarchy and the overall hierarchical structure.

(4) HMCN-F and HMCN-R are generally better than HMC-LMLP and Clus-HMC. This once again implies that it is effective for the HTMC task by integrating the predictions of each level in the category hierarchy and the overall hierarchical structure.

In summary, these evidences all indicate that the dependencies among different levels of the hierarchical structure and text-category associations are important for classifying documents in HMTC task. Moreover, these clues also reveal that HARNN can classify the documents into the multiple categories more effectively by integrating the predictions of each level in the category hierarchy and the overall hierarchical structure.

5.5.2 ***Performance on different levels (RQ2).*** In HMTC task, it is important to predict all categories in the entire hierarchy, and annotating categories of each level precisely is also critical. Thus, we conduct an experiment for comparing HARNN with HMC-LMLP, HMCN-F and HMCN-R on each level of the hierarchical structure separately. From Figure 6, we can see that HARNN outperforms all the other methods on all the category levels of the hierarchical structure, since HARNN leverages the information of hierarchical structure. Moreover, we also notice that the performance of all models tend to decrease when the hierarchy deepens, and HARNN retains superior performance compared to the other baselines. That is, as the hierarchical level increases, the categories of the level increase rapidly (e.g., the Patent dataset has 661, 8364 categories in level-3 and level-4 respectively, as shown in Table 1), which influences the model performance a lot. And HARNN considers the associations between texts and the hierarchical structure and dependencies among different levels of the hierarchical structure while the other three baselines do not. These results once again indicate that HARNN is more effective and powerful for HMTC task by considering both the associations between texts and the hierarchical structure and modeling the dependencies between different levels of the hierarchical structure.

5.5.3 ***Performance with different α (RQ3).*** In our HARNN model, the trade-off parameter α plays a crucial role which balances the importance of the local prediction and global prediction in Eq. (13). When α is smaller, the network tends to prioritize local predictions during learning. Conversely, as α is larger, the network is allowed to focus on global prediction more easily. We conduct an experiment by assigning different α from set $\{0, 0.1, 0.2, \ldots, 1.0\}$. As shown in Figure 7, as α increases, the performance of HARNN increases at the beginning, but it decreases afterwards both in two datasets. These results indicate that combining the local and global predictions properly is vital for achieving more accurate classification performance.

5.5.4 ***Hierarchical Attention Visualization (RQ4).*** An important characteristic of HARNN is that it can capture hierarchical attention information between text semantic representations and the hierarchical structure through visual attention vector $\widetilde{K}^h$ in Eq. (10) for each category level. The heatmap of the learned attention weights for a patent document is illustrated in Figure 8, where the document sample is the one in Figure 1. Please note that, we only show the part of the document text for illustration purposes. The color in Figure 8 changes from white to red while the value of attention weights increases. From Figure 8, we find that the HARNN model majorly learns to capture some key words in the text. For instance, the attention weights of the words "nuclear" and "plant" are obviously higher than the other words since these words dominate the prediction of the category *Physics* in Level-1. And we also observe that the attention weights of the words "nuclear" and "reactor" gradually increase when the hierarchy deepens, and meanwhile, the attention weights of the words "plant" and "injection" decrease. That is, the key words in different levels might be different due to the different concept description. Specifically, the words "nuclear" and "reactor" are more appropriate for describing the concepts of subsequent levels (e.g., the *Nuclear Reactors* concept in level-3), while the words "plant" and "injection" seem less relevant. That once again indicates that HARNN is capable of understanding the concept of a document text from shallow to deep properly. These results imply that HARNN can provide a good way to capture the hierarchical attention information in a given document via the HAM unit.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a model named HARNN for classifying documents into the most relevant categories level by level via combing the text and hierarchical structure. Specifically, we first applied a Documentation Representing Layer for obtaining the semantic encodings of text and categories. Then, we devised a Hierarchical Attention-based Recurrent Layer to model the dependencies between different levels by capturing the associations among text and hierarchical structure in a top-down fashion. After that, we designed a hybrid approach which is capable of predicting categories of each level while classifying the all categories in the entire hierarchy precisely. Finally, extensive experiments on two real-world datasets clearly demonstrated the effectiveness and explanatory power of HARNN.

In the future, there are still some directions for exploration. First, we would like to leverage the explicit constraints of the hierarchical structure (e.g., the hierarchical violation [28]) as regularization to improve classification performance. Second, we would like to address the issue of incomplete labels [35] and use more effective embedding for representing the hierarchical structure [21]. Third, as HAM is LSTM-like structure for encoding the category hierarchy information, so we will also try to design Bi-HAM (like Bi-LSTM) to capture the better hierarchy information from upward and downward simultaneously with our HARNN. Finally, as our HARNN is a general framework, we will test its performance on the other domains, such as the *protein function prediction* in biochemistry [7].

7 ACKNOWLEDGEMENTS

This research was partially supported by grants from the National Natural Science Foundation of China (Grants No. U1605251, 61727809, 61922073, 61602405), and the Young Elite Scientist Sponsorship Program of CAST.

REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] P. N. Bennett and N. Nguyen. Refined experts: improving classification in large taxonomies. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 11–18. ACM, 2009.

[3] W. Bi and J. T. Kwok. Multi-label classification on tree-and dag-structured hierarchies. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 17–24, 2011.

[4] H. B. Borges and J. C. Nievola. Multi-label hierarchical classification using a competitive neural network for protein function prediction. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE, 2012.

[5] A. Braytee, W. Liu, D. R. Catchpoole, and P. J. Kennedy. Multi-label feature selection using correlation information. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1649–1656. ACM, 2017.

[6] L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 78–87. ACM, 2004.

[7] R. Cerri, R. C. Barros, A. C. de Carvalho, and Y. Jin. Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC bioinformatics*, 17(1):373, 2016.

[8] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, 7(Jan):31–54, 2006.

[9] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[10] O. Dekel, J. Keshet, and Y. Singer. Large margin hierarchical classification. In *Proceedings of the twenty-first international conference on Machine learning*, page 27. ACM, 2004.

[11] A. Esuli, T. Fagni, and F. Sebastiani. Boosting multi-label hierarchical text categorization. *Information Retrieval*, 11(4):287–313, 2008.

[12] C. J. Fall, A. Törcsvári, K. Benzineb, and G. Karetka. Automated categorization in the international patent classification. In *Acm Sigir Forum*, volume 37, pages 10–25. ACM, 2003.

[13] E. Gibaja and S. Ventura. Multi-label learning: a review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(6):411–444, 2014.

[14] J. C. Gomez and M.-F. Moens. A survey of automated hierarchical classification of patents. In *Professional Search in the Modern World*, pages 215–249. Springer, 2014.

[15] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[16] J. Han, C. Wang, and A. El-Kishky. Bringing structure to text: mining phrases, entities, topics, and hierarchies. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1968–1968. ACM, 2014.

[17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[18] Z. Huang, Q. Liu, E. Chen, H. Zhao, M. Gao, S. Wei, Y. Su, and G. Hu. Question difficulty prediction for reading problems in standard tests. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[21] J. Ma, P. Cui, X. Wang, and W. Zhu. Hierarchical taxonomy aware network embedding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1920–1929. ACM, 2018.

[22] A. Mayne and R. Perry. Hierarchically classifying documents with multiple labels. In *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*, pages 133–139. IEEE, 2009.

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[24] Z. Ren, M.-H. Peetz, S. Liang, W. Van Dolen, and M. De Rijke. Hierarchical multi-label classification of social text streams. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 213–222. ACM, 2014.

[25] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Learning hierarchical multi-category text classification models. In *Proceedings of the 22nd international conference on Machine learning*, pages 744–751. ACM, 2005.

[26] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7(Jul):1601–1626, 2006.

[27] M. E. Ruiz and P. Srinivasan. Hierarchical text categorization using neural networks. *Information Retrieval*, 5(1):87–118, 2002.

[28] C. N. Silla and A. A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72, 2011.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[30] H. Tao, S. Tong, H. Zhao, T. Xu, B. Jin, and Q. Liu. A radical-aware attention-based model for chinese text classification. In *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[31] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel. Decision trees for hierarchical multi-label classification. *Machine learning*, 73(2):185, 2008.

[32] X. Wang and G. Sukthankar. Multi-label relational neighbor classification using social context features. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 464–472. ACM, 2013.

[33] J. Wehrmann, R. Cerri, and R. Barros. Hierarchical multi-label classification networks. In *International Conference on Machine Learning*, pages 5225–5234, 2018.

[34] F. Wu, J. Zhang, and V. Honavar. Learning classifiers using hierarchically structured class taxonomies. In *International Symposium on Abstraction, Reformulation, and Approximation*, pages 313–320. Springer, 2005.

[35] L. Xu, Z. Wang, Z. Shen, Y. Wang, and E. Chen. Learning low-rank label correlations for multi-label classification with missing labels. In *2014 IEEE International Conference on Data Mining*, pages 1067–1072. IEEE, 2014.

[36] B. Yang, J.-T. Sun, T. Wang, and Z. Chen. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 917–926. ACM, 2009.

[37] L. Zhang, K. Xiao, Q. Liu, Y. Tao, and Y. Deng. Modeling social attention for stock analysis: An influence propagation perspective. In *2015 IEEE International Conference on Data Mining*, pages 609–618. IEEE, 2015.